

CTF as a possible BTF data source

Friday 28 August 2020 07:45 (45 minutes)

Last year we introduced support for the Compact C Type Format (CTF) into the GNU toolchain. We have since improved the linking of CTF so that types are properly deduplicated: the work is done by libctf on ld's behalf so that other programs can do what ld does. With the aid of a few dozen lines of makefile changes and a 300-odd line program using libctf, we can now produce a fully deduplicated description of all types in the kernel, with types specific to single modules localized appropriately. A recent kernel (with a 3000-module enterprise configuration) comes to about 7MiB of types, after compression, of which about half is core-kernel stuff and the rest are types only used by single modules (that users can often avoid loading).

We plan to do more changes to improve CTF in ways the kernel team might find useful (representing static functions' types is planned, as well as further space reductions), but I don't want to make this up entirely on my own, so I thought I should ask what people need.

One obviously essential piece not present yet is turning CTF into BTF in the first place. Directly translating CTF into BTF and vice versa as I proposed last year is possible, but BTF is such a moving target that I fear we might have trouble keeping up (we can hardly release binutils as often as the kernel is released, and nobody upgrades the two in sync anyway).

But bidirectional conversion straight from CTF<->BTF might not actually be necessary for the kernel to exploit CTF: emitting C source code corresponding to CTF is definitely possible, and this might be just as useful: this is already doable with BTF, of course, so this might serve as a bidirectional gateway that requires less chasing. At the very least going from CTF -> BTF rather than from DWARF -> BTF would speed up compiles and make them take much less disk space (a recent test using an enterprise kernel showed a space saving by generating CTF instead of DWARF of around ten gigabytes). But there could be other advantages, too (among other things CTF is much easier to change than DWARF at present).

Does anyone have any other ideas of things I might do to make your lives easier? A CTF file format bump is happening in the near future, so now is the time to propose new stuff. I want to take some of the burden of the more boring parts of BTF off you and drop it into binutils where you can forget about it, if possible.

I agree to abide by the anti-harassment policy

I agree

Primary author: ALCOCK, Nick (Oracle Corporation)

Presenter: ALCOCK, Nick (Oracle Corporation)

Session Classification: GNU Toolchain MC

Track Classification: GNU Toolchain MC